D 16

```
FFFFFFFFFFFFFFF        111               111           AAAAAAAAA
FFFFFFFFFFFFFFF        111               111           AAAAAAAAA
FFFFFFFFFFFFFFF        111               111           AAAAAAAAA
FFF                 111111            111111        AAA         AAA
FFF                 111111            111111        AAA         AAA
FFF                 111111            111111        AAA         AAA
FFF                    111               111        AAA         AAA
FFF                    111               111        AAA         AAA
FFF                    111               111        AAA         AAA
FFFFFFFFFFFF           111               111        AAA         AAA
FFFFFFFFFFFF           111               111        AAA         AAA
FFFFFFFFFFFF           111               111        AAA         AAA
FFF                    111               111        AAAAAAAAAAAAAAA
FFF                    111               111        AAAAAAAAAAAAAAA
FFF                    111               111        AAAAAAAAAAAAAAA
FFF                    111               111        AAA         AAA
FFF                    111               111        AAA         AAA
FFF                    111               111        AAA         AAA
FFF                 111111111         111111111     AAA         AAA
FFF                 111111111         111111111     AAA         AAA
FFF                 111111111         111111111     AAA         AAA
```

D 15

```
**FILE**ID**IODONE
```

```
  IIIIII      OOOOOO    DDDDDDDD      OOOOOO    NN      NN  EEEEEEEEEE
  IIIIII      OOOOOO    DDDDDDDD      OOOOOO    NN      NN  EEEEEEEEEE
    II      OO      OO  DD      DD  OO      OO  NN      NN  EE
    II      OO      OO  DD      DD  OO      OO  NN      NN  EE
    II      OO      OO  DD      DD  OO      OO  NNNN    NN  EE
    II      OO      OO  DD      DD  OO      OO  NNNN    NN  EE
    II      OO      OO  DD      DD  OO      OO  NN  NN  NN  EEEEEEEE
    II      OO      OO  DD      DD  OO      OO  NN  NN  NN  EEEEEEEE
    II      OO      OO  DD      DD  OO      OO  NN    NNNN  EE
    II      OO      OO  DD      DD  OO      OO  NN    NNNN  EE
    II      OO      OO  DD      DD  OO      OO  NN      NN  EE              ....
    II      OO      OC  DD      DD  OO      OO  NN      NN  EE              ....
  IIIIII      OOOOOO    DDDDDDDD      OOOOOO    NN      NN  EEEEEEEEEE      ....
  IIIIII      OOOOOO    DDDDDDDD      OOOOOO    NN      NN  EEEEEEEEEE      ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000            .TITLE  IODONE - POST REQUEST DONE TO USER
0000            .IDENT  'V04-000'
0000
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER  *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000
0000    ;++
0000    ;
0000    ; FACILITY:  F11ACP STRUCTURE LEVEL 1
0000    ;
0000    ; ABSTRACT:
0000    ;
0000    ;       THIS ROUTINE POSTS I/O COMPLETION FOR THE INDICATED FCP REQUEST.
0000    ;
0000    ; ENVIRONMENT:
0000    ;
0000    ;       STARLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
0000    ;       AND INTERNAL EXEC ROUTINES. THIS ROUTINE MUST BE CALLED IN
0000    ;       KERNEL MODE.
0000    ;
0000    ;--
0000    ;
0000    ; AUTHOR:  ANDREW C. GOLDSTEIN, CREATION DATE:  20-DEC-1976  11:25
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       V03-002 HH0051          Hai Huang               21-Aug-1984
0000    ;               Call check_dismount before posting I/O completion.
0000    ;
0000    ;       V03-001 ACG0320         Andrew C. Goldstein,    22-Mar-1983  12:25
0000    ;               Change byte count handling to track IOPOST
0000    ;
0000    ;       V02-001 LJK0076         Lawrence J. Kenah       3-Nov-1981
0000    ;               Remove check for "queue previously not empty" when making
0000    ;               software interrupt request. The request is always made.
0000    ;
```

```
                 0000   ;**
                 0000
                 0000   ;
                 0000   ; EQUATED SYMBOLS:
                 0000   ;
        00000004 0000   PACKET  =4                                      ; ADDRESS OF I/O PACKET ARG
                 0000
                 0000           $ABDDEF                                 ; DEFINE BUFFER PACKET OFFSETS
                 0000           $FIBDEF                                 ; DEFINE FIB OFFSETS
                 0000           $IRPDEF                                 ; DEFINE I/O PACKET OFFSETS
                 0000           $UCBDEF                                 ; DEFINE UCB OFFSETS
                 0000           $VCBDEF                                 ; DEFINE VCB OFFSETS
                 0000           $IPLDEF                                 ; DEFINE IPL SYMBOLS
                 0000           $PRDEF                                  ; DEFINE PRIORITY LEVELS
                 0000           $IODEF                                  ; DEFINE I/O FUNCTION CODES
```

```
                         0000  ;++
                         0000  ;
                         0000  ; FUNCTIONAL DESCRIPTION:
                         0000  ;
                         0000  ;       THIS ROUTINE POSTS I/O COMPLETION FOR THE INDICATED FCP REQUEST.
                         0000  ;
                         0000  ; CALLING SEQUENCE:
                         0000  ;       CALL    IODONE (ARG1)
                         0000  ;
                         0000  ; INPUT PARAMETERS:
                         0000  ;       ARG1: ADDRESS OF I/O PACKET
                         0000  ;
                         0000  ; IMPLICIT INPUTS:
                         0000  ;       USER_STATUS: STATUS OF I/O REQUEST
                         0000  ;
                         0000  ; OUTPUT PARAMETERS:
                         0000  ;       NONE
                         0000  ;
                         0000  ; IMPLICIT OUTPUTS:
                         0000  ;       IOC$GL_PSBL: TAIL OF I/O POST QUEUE
                         0000  ;
                         0000  ; ROUTINE VALUE:
                         0000  ;       NONE
                         0000  ;
                         0000  ; SIDE EFFECTS:
                         0000  ;       I/O PACKET PLACED ON I/O POST QUEUE
                         0000  ;       VOLUME CHECKED FOR DISMOUNT
                         0000  ;
                         0000  ;--
                         0000
                     00000000          .PSECT  $CODE$,NOWRT,LONG
                         0000
                         0000  IO_DONE::
                   00FC  0000          .WORD   ^M<R2,R3,R4,R5,R6,R7>    ; SAVE REGISTERS
        56    04 AC   D0 0002          MOVL    PACKET(AP),R6            ; GET PACKET ADDRESS
    38 A6  0000'CF    7D 0006          MOVQ    W^USER_STATUS,IRP$L_MEDIA(R6) ; SET STATUS IN PACKET
        06    00      EF 000C          EXTZV   #IRP$V_FCODE,#IRP$S_FCODE,-
        57    20 A6      000F                  IRP$W_FUNC(R6),R7        ; GET FUNCTION CODE WITHOUT QUALIFIERS
        0C    57      91 0012          CMPB    R7,#IO$_READPBLK         ; IF READ PHYSICAL
              3A      13 0015          BEQL    10$
        0B    57      91 0017          CMPB    R7,#IO$_WRITEPBLK        ; OR WRITE DO SPECIAL PROCESSING
              35      13 001A          BEQL    10$
                         001C  ;
                         001C  ; POST PROCESSING FOR ALL ACP FUNCTIONS: BUMP DOWN THE VOLUME TRANSACTION
                         001C  ; COUNT AND DO THE FIXUPS FOR THE BUFFER PACKET.
                         001C  ;
        54 0000'CF    D0 001C          MOVL    W^CURRENT_VCB,R4         ; GET VCB ADDRESS
           0C A4      B7 0021          DECW    VCB$W_TRANS(R4)          ; DEDUCT THIS REQ FROM TRANS COUNT
  2C 2A A6    03      E1 0024          BBC     #IRP$V_COMPLX,IRP$W_STS(R6),30$ ; BRANCH IF NO BUFFER PACKET
        54    2C B6   D0 0029          MOVL    @IRP$L_SVAPTE(R6),R4     ; GET BUFFER DESCRIPTOR ADDRESS
              12 A4   B4 002D          CLRW    <ABD$C_NAME*ABD$C_LENGTH>+ABD$W_COUNT(R4)
                         0030                                           ; INHIBIT WRITE-BACK OF NAME STRING
        52    08 A4   9E 0030          MOVAB   <ABD$C_FIB*ABD$C_LENGTH>+ABD$W_TEXT(R4),R2
        53    62      3C 0034          MOVZWL  (R2),R3                  ; GET OFFSET ADDRESS OF FIB IN PACKET
        52    53      C0 0037          ADDL    R3,R2                    ; COMPUTE ABSOLUTE ADDRESS
00  0000'CF  0040 8F  2C 003A          MOVC5   #FIB$C_LENGTH,W^LOCAL_FIB,#0,-
     01 A2    0A A4      0042                  <ABD$C_FIB*ABD$C_LENGTH>+ABD$W_COUNT(R4),1(R2)
```

```
                              0046                                        ; COPY LOCAL FIB BACK INTO PACKET
   0A 2A A6     01      E2    0046          BBSS    #IRP$V_FUNC,IRP$W_STS(R6),30$ ; IF READ BIT IS SET, KEEP
      32 A6     05      B0    004B          MOVW    #ABD$C_ATTRIB,IRP$W_BCNT(R6) ; ELSE DUMP ATTRIBUTE TEXT
                04      11    004F          BRB     30$
                              0051  ;
                              0051  ; FOR READ/WRITE PHYSICAL, KNOCK DOWN THE VIRTUAL BIT IN THE PACKET. ONLY
                              0051  ; ERRORS COME THROUGH HERE, AND WE DON'T WANT TO SEE THEM AGAIN (I/O POST
                              0051  ; RECYCLES VIRTUAL I/O ERRORS FOR ACP ERROR PROCESSING).
                              0051  ;
                              0051          ASSUME  IRP$V_VIRTUAL LE 7
      2A A6     10      8A    0051  10$:    BICB    #IRP$M_VIRTUAL,IRP$W_STS(R6) ; CLEAR THE VIRTUAL BIT
                              0055
    0000'CF     00      FB    0055  30$:    CALLS   #0,W^CHECK_DISMOUNT     ; CHECK THE VOLUME FOR DISMOUNT
                              005A
00000000'FF     66      0E    005A          INSQUE  (R6),@IOC$GL_PSBL       ; INSERT PACKET INTO QUEUE
                              0061          SOFTINT #IPL$_IOPOST            ; SIGNAL I/O POST INTERRUPT
                        04    0064          RET
                              0065
                              0065
                              0065
                              0065          .END
```

```
LO
Sy

AQ
BI
DI
FC
HE
IN
LC
LC
LD
LD
LO
LO
LO
MV
RV
SY
VC
WC



PS
--
.
$L
$L
$L
$L
$L
$C



Ph
--
In
Co
Pa
Sy
Pa
Sy
Ps
Cr
As

Th
25
Th
22
6
```

```
ABD$C_ATTRIB                     = 00000005
ABD$C_FIB                        = 00000001
ABD$C_LENGTH                     = 00000008
ABD$C_NAME                       = 00000002
ABD$W_COUNT                      = 00000002
ABD$W_TEXT                       = 00000000
AQB_TYPE                         = 00000005
BITMAP_TYPE                      = 00000001
CHECK_DISMOUNT                     ********   X   02
CURRENT_VCB                        ********   X   02
DIRECTORY_TYPE                   = 00000002
FCB_TYPE                         = 00000000
FIB$C_LENGTH                     = 00000040
HEADER_TYPE                      = 00000000
INDEX_TYPE                       = 00000003
IO$_READPBLK                     = 0000000C
IO$_WRITEPBLK                    = 0000000B
IOC$GL_PSBL                        ********   X   02
IO_DONE                            00000000 RG    02
IPL$_IOPOST                      = 00000004
IRP$L_MEDIA                      = 00000038
IRP$L_SVAPTE                     = 0000002C
IRP$M_VIRTUAL                    = 00000010
IRP$S_FCODE                      = 00000006
IRP$V_COMPLX                     = 00000003
IRP$V_FCODE                      = 00000000
IRP$V_FUNC                       = 00000001
IRP$V_VIRTUAL                    = 00000004
IRP$W_BCNT                       = 00000032
IRP$W_FUNC                       = 00000020
IRP$W_STS                        = 0000002A
LOCAL_FIB                          ********   X   02
MVL_TYPE                         = 00000004
PACKET                           = 00000004
PR$_SIRR                         = 00000014
RVT_TYPE                         = 00000003
USER_STATUS                        ********   X   02
VCB$W_TRANS                      = 0000000C
VCB_TYPE                         = 00000002
WCB_TYPE                         = 00000001

                                              +----------------+
                                              ! Psect synopsis !
                                              +----------------+

PSECT name                       Allocation          PSECT No.  Attributes
----------                       ----------          ---------  ----------
.  ABS  .                        00000000  (    0.)  00 (  0.)  NOPIC    USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                            00000000  (    0.)  01 (  1.)  NOPIC    USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
$CODE$                           00000065  (  101.)  02 (  2.)  NOPIC    USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
```

```
                                               +-------------------------+
                                               ! Performance indicators !
                                               +-------------------------+

Phase                        Page faults    CPU Time        Elapsed Time
-----                        -----------    --------        ------------
Initialization                        31    00:00:00.09     00:00:00.52
Command processing                   148    00:00:00.76     00:00:04.45
Pass 1                               340    00:00:11.62     00:00:29.59
Symbol table sort                      0    00:00:02.09     00:00:03.19
Pass 2                                45    00:00:01.88     00:00:04.49
Symbol table output                    6    00:00:00.09     00:00:00.13
Psect synopsis output                  2    00:00:00.02     00:00:00.02
Cross-reference output                 0    00:00:00.00     00:00:00.00
Assembler run totals                 575    00:00:16.55     00:00:42.46

The working set limit was 1200 pages.
64141 bytes (126 pages) of virtual memory were used to buffer the intermediate code.
There were 70 pages of symbol table space allocated to hold 1336 non-local and 2 local symbols.
254 source lines were read in Pass 1, producing 13 object records in Pass 2.
21 pages of virtual memory were used to define 20 macros.

                                              +----------------------------+
                                              ! Macro library statistics !
                                              +----------------------------+

Macro library name                            Macros defined
------------------                            --------------
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                          6
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       7
TOTALS (all libraries)                                 13

1415 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:IODONE/OBJ=OBJ$:IODONE MSRC$:FCPPRE/UPDATE=(ENH$:FCPPRE)+MSRC$:IODONE/UPDATE=(ENH$:IODONE)+EXECML$/LIB
```

0165 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY